Computational Analysis of a Corpus of Poems

Blair Drummond, Supervisors: Dr. Diana Inkpen, Dr. Chris Tanasescu

School of Electrical Engineering and Computer Science, University of Ottawa

The Goal

Does our program understand poetry?
No. Not at all, but in the future, it could help researchers across many disciplines learn a bit more about what is happening in poetry.

The program developed here uses a collection of tools to find interesting data about poems, conducting a linguistic analysis in order to mine information about a few features. What we are looking for:

-What do sentences tend to look like in poetry

-Where, and how often, do poets use line breaks?

And what is the relationship between them and the syntax, and the meaning(s) constructed in the poem?

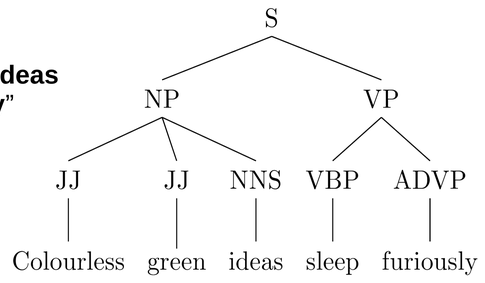
Once we have this tool built, we can gain more insight into how these patterns appear (over a very large scale). Charting the similarities between poets, and understanding larger relationships in the field as a graph. And also allowing for a better understanding of the syntactic differences between poetry and normal writing.

Hold on. What is Syntax?

Syntax is about form. There are parallels to math, but in math the form is explicit; for instance, In “(2+2)*5”, the brackets don't contribute any *meaning*, but they change the way we interpret the expression. The same is true in language... except that people have to judge where the metaphorical brackets are.

“**Colourless green ideas sleep furiously**”

(Chomsky, 1955)



Also like math, the structure and the content are independent. This sentence doesn't make sense, but it is perfectly interpretable, because it is well formed. In fact, none of the words, themselves, in the sentence matter, only their category is important. However, not all random sequences of words can have a structure. This 'sentence' for instance, is beyond incoherent.

*“**Cigarettes respectable battles greasy save.**”*
(Pinker, 1994)

What makes one sentence grammatical, and one ungrammatical, is one of the fields of study in linguistics, but that is not relevant to our discussion here. Our purpose here is to understand *How* poets choose to use sentence structures and form-

And when they decide to ignore them.

What We Are Measuring

In many poems, even fundamentals such as English's SUBJECT, VERB, OBJECT order are violated. For instance, Samuel Coleridge's poem, “Kubla Khan”. \

“In Xanadu did Kubla Khan
A stately pleasure-dome decree:
Where Alph, the sacred river, ran
Through caverns measureless to man
Down to a sunless sea...”

So these sorts of sentences are important because they push the envelope of what a grammatical sentence is, and they have a very noticeable effect.

Line breaks and stanza breaks also shake the text, but visually, by making a more jarring transition for the reader.

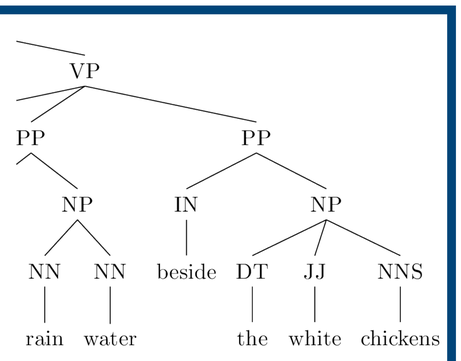
In this case, the poet breaks the sentence across many stanzas. How many times this occurs is important, but we can also ask if there is a particular syntactic position where poets are more likely to do this. By combining our parse trees with the line breaks in the poems, we can get the numbers on how often, and where this occurs. After that, we can get a picture of the overall trends.

“so much depends upon

a red wheel barrow

glazed with rain water

beside the white chickens”



(William Carlos Williams, The Red Wheelbarrow)

The Software

The software we have built relies on existing material, using the Stanford CoreNLP system (which is written in Java) to do the extraction of the parse trees, but much of it was built from scratch, mostly using Bash, and Python, distributed across several files which run in a pipeline.

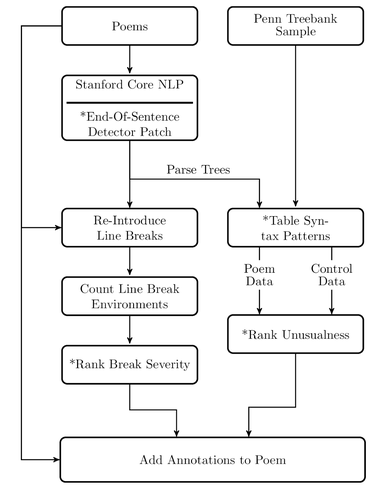
The System has two input sources, a corpus of about 10 000 poems accessed from the internet, and also a large sample of training text from Penn TreeBank data. The Penn Data is a collection of many written texts, and it's being used as a benchmark for a 'normal' sentence, so that we can rank unusualness in a poem based on how it deviates from the control data. In the end, the system breaks down into 3 logical pieces: Get the parse trees, extract the relevant data from the parses, and then analyze the data by comparing it to the control data.

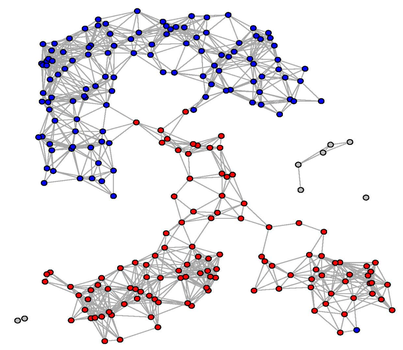
So far, we have successfully set up most of the data extraction. The first step was adjusting the Stanford tool to properly tag words with case in poems. Normally, case is an important indicator of proper nouns, but in 'Kubla Khan', for instance, the beginning of every line is uppercase, and so this heuristic had to be ignored.

After the parse trees are produced, the line breaks and stanza break information is destroyed, and so that data has to be reconstructed from the old document.

The Software: continued

The problem was not quite trivial, because the Stanford tool muddled the relationship between the poem and the parse tree, for instance, it split some contractions into two tokens. “Don't” becomes “Do” and “n't”, because they are, logically, two different words. This is really a remarkable feat in many ways, but it makes matching tokens a bit tricky. We integrated the line breaks by taking a line from the poem, identifying special cases such as contractions, modifying it into a regular expression, and using it to find that line in the parse tree. After that, the task of gathering all the local environments for the line breaks was a fairly simple task.





Future Work

1. There is one limitation damaging current output. An unfortunate shortcoming of the Stanford CoreNLP tool, it doesn't have any tool for fuzzy detection of sentence boundaries. So for instance, if you took William Carlos William's poem (on the left), and continued it, the Stanford tool would parse it as a single sentence (because there is no punctuation). So the next phase of the development is to develop a language model which can make these parses accurate.

2. Deciding how to measure 'unusualness' is not straightforward. In linguistics, the distributions tend to be very messy. In any given sample (almost regardless of the size) there are going to be valid grammatical structures which do not appear at all, but may be perfectly valid. In fact, many of the structures which don't appear may be perfectly usual sounding. So there has to be a much more sophisticated way of weighting sentences than simply comparing it to an occurrences-over-total metric. At the moment we are looking at modelling unusualness with a tree-adjoining grammar. By creating a narrowly defined grammar, the hope is that we can model what sentences are generated by a 'usual' formal language.

GitHub

You can access the code for the project at
https://github.com/blairdrummond/NLP-Poetry

For more on the Stanford CoreNLP tool see their website
http://nlp.stanford.edu/software/corenlp.shtml

Acknowledgements

This project was made possible thanks to the UROP grant from University of Ottawa, and an SSHRC research grant for the project “Poetry Computational Graphs”, for which Dr. Tanasescu is Principal Investigator and Dr. Inkpen is Co-Investigator.

Special Thanks to Dr. Inkpen and Dr. Tanasescu for supervising the project, and thanks to the Stanford NLP group who created the Open-Source software that we used.